1) Digit extraction – To get last digit we do modulo 10
E.g – 721 last digit is 1 . 721 % 10 = 1 then we can divide number by 10 to get next digit.
TC – $O(\log_{10} N)$ TC is log of N to the base 10 because each time we divide by 10.

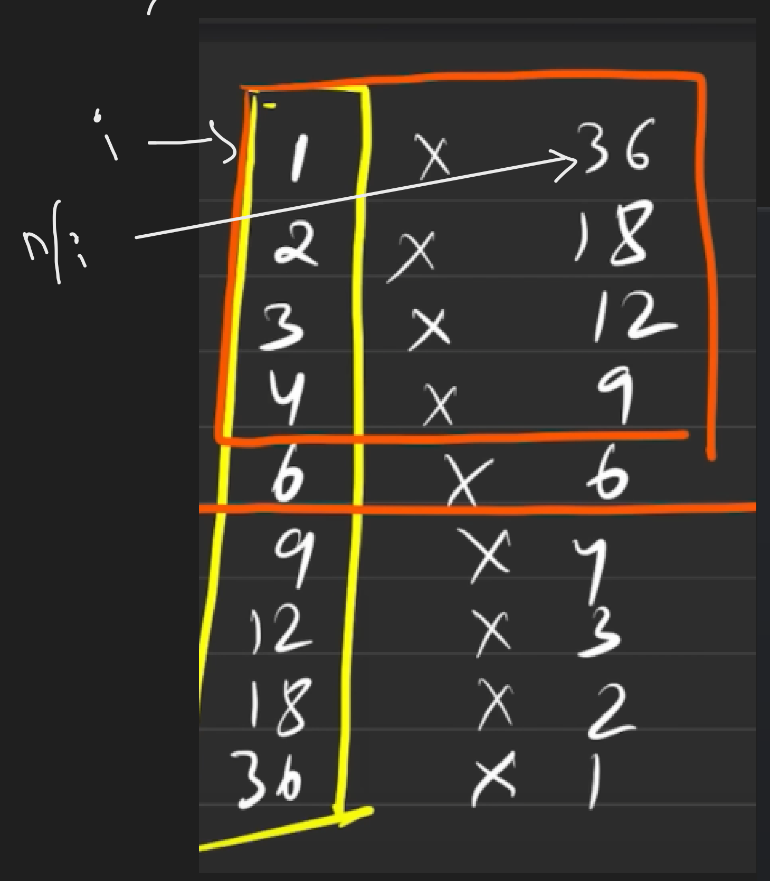
2) Reverse number – Almost similar to above.

3) Palindrome – Reverse of number is same as original.
Find reverse & compare with given number.

4) Armstrong number – Count of number of digits in number as power of each digit and their sum.

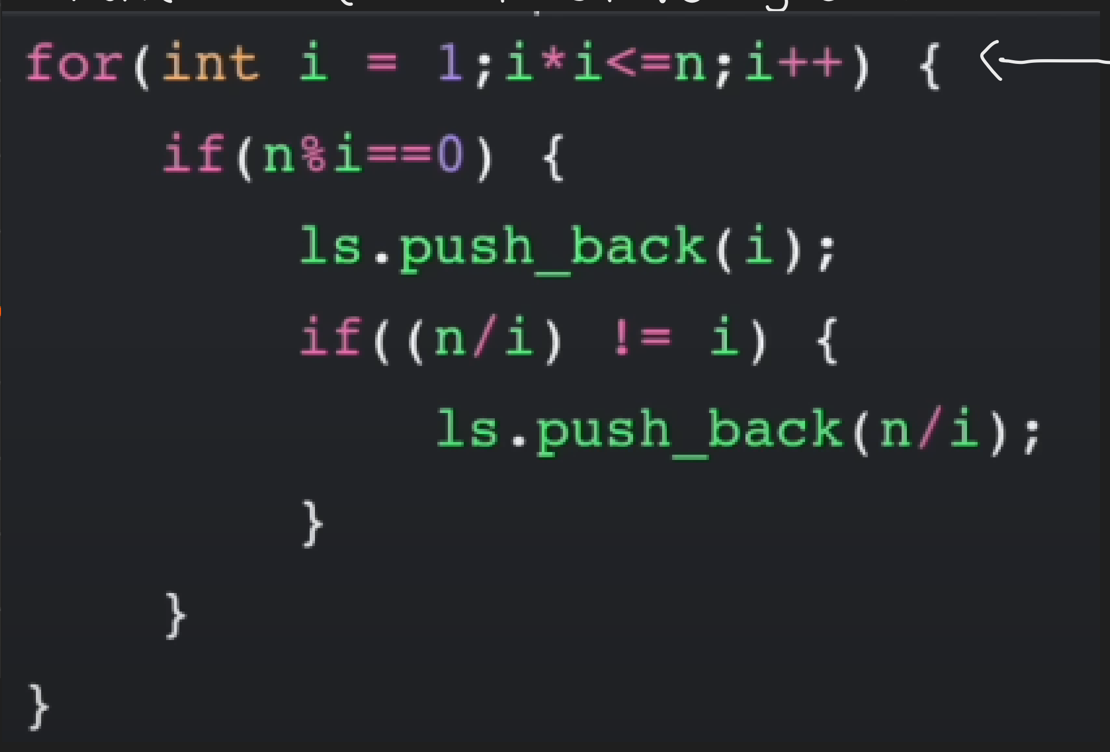
Like $371 = 3^3 + 7^3 + 1^3 = 371$

5) Print all divisors = Divisor is numbers which completely divide n (remainder zero).



1	X	36
2	X	18
3	X	12
4	X	9
6	X	6
9	X	4
12	X	3
18	X	2
36	X	1

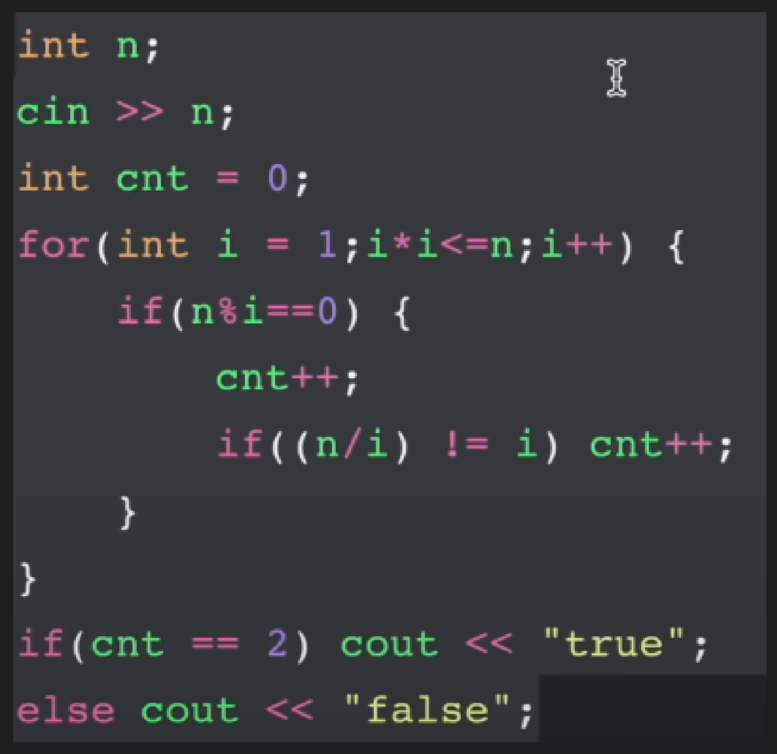
We can simply loop till $\sqrt{n}$ and take i and n/i and we will have all the factors. Just make sure i & n/i are not same.



```
for(int i = 1;i*i<=n;i++) {
    if(n%i==0) {
        ls.push_back(i);
        if((n/i) != i) {
            ls.push_back(n/i);
        }
    }
}
```

← O(sqrt(n))

6) Prime numbers – A number that is divisible by 1 and itself (1 is not a prime number)



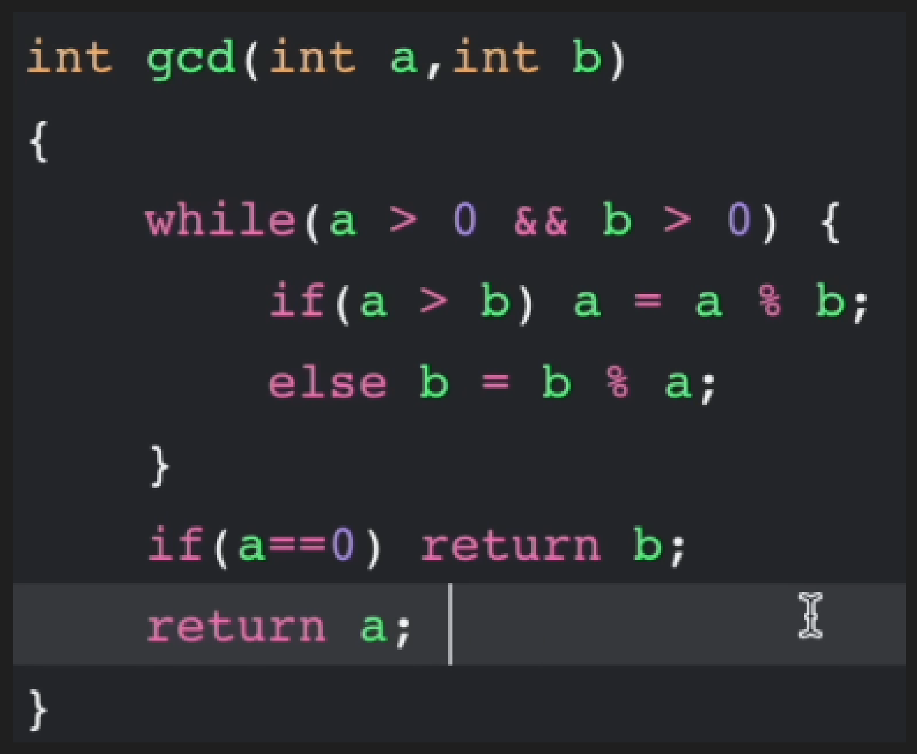
```
int n;
cin >> n;
int cnt = 0;
for(int i = 1;i*i<=n;i++) {
    if(n%i==0) {
        cnt++;
        if((n/i) != i) cnt++;
    }
}
if(cnt == 2) cout << "true";
else cout << "false";
```

We can find count of all factors of number and if their count exceeds 2 then it's not a prime number.

7) GCD / HCF – Greatest Common Divisor OR Highest Common Factor between two numbers is the largest common factor between the two which can divide both numbers
GCD (20,40) → 20.
Any two numbers always have a G.C.D of 1 at least.

A brute force approach would be to go from i = 1 to i = min(n₁, n₂) & if both are divisible by i replace result with i.

We can also solve this problem using euclidean algorithm. We find gcd(a-b, b) where a>b till one of them is zero and return the number that's not zero.
TC – $O(\log_\phi(\min(a,b)))$
↑
This value keeps fluctuating depending on examples.



```
int gcd(int a,int b)
{
    while(a > 0 && b > 0) {
        if(a > b) a = a % b;
        else b = b % a;
    }
    if(a==0) return b;
    return a;
}
```

8) LCM – Lowest Common Multiple is the smallest value which can be divided by both n₁ & n₂

LCM = (num1 * num2) / GCD(num1, num2)